ppm

Srini, rs@toprllc.com

August 24, 2021

1 Introduction

This projectlet is a companion to the password generator. A method of recording the passwords and retrieving it on demand is the key feature developed.

Passwords are organized by a context - typically a service or website such as **www.google.com** or **www.amazon.com**, a username for the site.

Storage of the passwords in files is of course a seriously risky proposition; so the file itself is designed to be **secure** with its own password protection.

2 User Needs

The application will be developed to meet the following needs:

Generate and store passwords The user needs a way to specify a context and generate a new password for the context.

Store passwords The passwords need to be recorded in a file (type **ppm**) which itself is password protected.

Show the password for a context Knowing the password to the **ppm** file, the application supports listing the passwords stored away.

3 Design

3.1 Libraries

A secure file facility and in particular a secure password file is designed to support this tool. Encryption, key derivation and other such building blocks are implemented using **OpenSSL** librarie (`https://wiki.openssl.org/index.php/Main_Page`).

3.2 Secure File Format

The password file is a secure file which starts with a header defined as:

```
type headerType is
record
 pwd : openssl.evp.digest.DigestValue(1..openssl.evp.digest.EVP_MAX_MD_SIZE)
   := (others => 0) ;
 iv : aliased openssl.evp.cipher.InitializationVector(1..openssl.evp.cipher.EVP_MAX_IV_
   := (others => 0) ;
 sig : openssl.evp.digest.DigestValue(1..openssl.evp.digest.EVP_MAX_MD_SIZE)
   := (others => 0) ;
end record ;
```

User supplied cleartext password is first transformed into a key using a key derivation function. So the access to the file is granted based on matching this password.

At the time of creation of the file, an initialization vector is created and is stored in the header of the file.

Finally a **message digest** of the entire file is generated and stored as part of the header.

The payload or the user file is encrypted and the resulting stream follow the header.

3.3 Password File

The password file itself is a simplistic comma separated file. Each line of this file is a triplet of context, username and the actual password.

4 Examples

4.1 Create a new password file

```
$ ../obj/ppm create --help
Usage: ppm.exe create a new personal password database

 -v, --verbose           be verbose
 -p, --password-file=ARG Password File
 -k, --keep              keep intermediate files
 -f, --force             force the creation. erase existing file
```

Example

```
$ ../obj/ppm create -p=newpwd.ppm -f
Password myname
```

4.2 Generate a password and save

```
$ ../obj/ppm set --help
Usage: ppm.exe set the credentials for a new context(site)

 -v, --verbose                be verbose
 -p, --password-file=ARG      Password File
 -k, --keep                   keep intermediate files
 -f, --word-list-file-name=ARG Word list file name
 -t, --separator=ARG          Separator
 -g, --generate               generate a password
 -b, --builtin-wordlist       use builtin wordlist for generation
 -o, --override               override the existing password
 -s, --segments=ARG           Number of Segments
 -m, --max-word-length=ARG    Maximum length of words
```

Example

```
$ ../obj/ppm set -p=newpwd.ppm -g -b -f hotmail myname
Password myname
$ ../obj/ppm show -p=newpwd.ppm hotmail .
Password myname
Context : hotmail , Username : myname , Password : pylorus10396Anole18213
```

4.3 Create a new database with the same contents

```
$ ../obj/ppm reset --help
Usage: ppm.exe reset the credentials db. create a new one with a different password
 -v, --verbose          be verbose
 -p, --password-file=ARG Password File
 -k, --keep             keep intermediate files
 -f, --force            force the creation. erase existing file
```

Example

```
$ ../obj/ppm reset -p=newpwd.ppm newpwdo.ppm
Password myname
New Password mynewname
$ ../obj/ppm show -p=newpwdo.ppm
Password mynewname
$ ../obj/ppm show -p=newpwdo.ppm . .
Password mynewname
Context : ctx , Username : uname , Password : pwd
Context : hotmail , Username : myname , Password : pylorus10396Anole18213
Context : myname , Username :  , Password : spine64441Microbe54261
```

5 Implementation

Ada bindings to the openssl library are maintained in a distinct library:
`https://gitlab.com/ada23/sslada.git`

The secure file format is maintained as part of the **pwdgen** projectlet:
`https://gitlab.com/ada23/pwdgen.git`

Utilizing the above, the secure password file and the application to password management as outlined is to be found in:
`https://gitlab.com/ada23/ppm.git`